

Capturing Lombardi Flow in Orthogonal Drawings by Minimizing the Number of Segments

Md. Jawaherul Alam, Michael Dillencourt, and Michael T. Goodrich

Department of Computer Science, University of California, Irvine, CA, USA

Abstract. Inspired by the artwork of Mark Lombardi, we study the problem of constructing orthogonal drawings where a small number of horizontal and vertical line segments covers all vertices. We study two problems on orthogonal drawings of planar graphs, one that minimizes the total number of line segments and another that minimizes the number of line segments that cover all the vertices. We show that the first problem can be solved by a non-trivial modification of the flow-network orthogonal bend-minimization algorithm of Tamassia, resulting in a polynomial-time algorithm. We show that the second problem is NP-hard even for planar graphs with maximum degree 3. Given this result, we then address this second optimization problem for trees and series-parallel graphs with maximum degree 3. For both graph classes, we give polynomial-time algorithms for upward orthogonal drawings with the minimum number of segments covering the vertices.

1 Introduction

The American artist Mark Lombardi [23] produced many hand-drawn visualizations of social networks to represent conspiracy theories, with the image shown in Fig. 1 being one of his most famous works. Many people have observed that his drawings have strong aesthetic qualities and high readability. As a result, there has been considerable work in the graph drawing literature inspired by his work (e.g., see [2, 8, 12, 13, 17, 31, 34]). For example, Duncan *et al.* [13] observed that in the art work by Lombardi edges tend to be nearly evenly spaced around each vertex, and they introduced the notion of a *Lombardi drawing*, where the edges incident on each vertex are drawn to have perfect angular resolution. Our approach in this paper, however, is to focus instead on a different aesthetic quality evident in some of the works of Lombardi.

Another aesthetic property prevalent in Lombardi's art work is that he tends to place many vertices on consecutive stretches of linear or circular segments that go across the whole drawings; see Fig. 1. This creates a metaphor of a "visual flow" across a drawing. Inspired by this property, we are interested in this paper in a new aesthetic criteria for orthogonal drawings of graphs. Namely, we would like to place the vertices of a graph on a small number of contiguous line segments in the drawing.

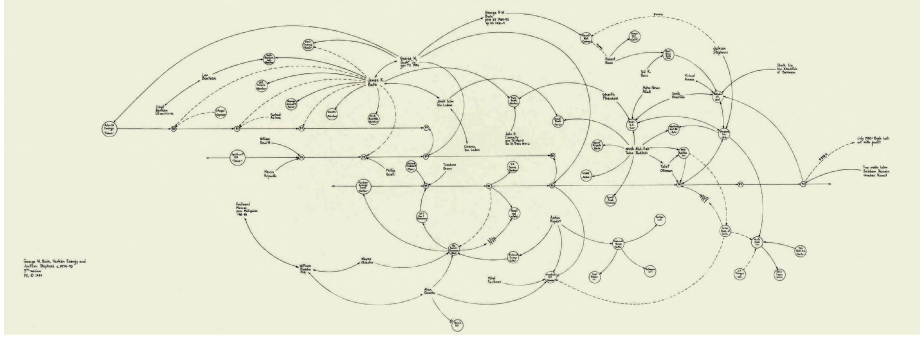


Fig. 1: Mark Lombardi: George W. Bush, Harken Energy, and Jackson Stevens c. 1979–90, 1999. Graphite on paper, 20×44 inches [23].

As is well-known in the graph drawing literature, *orthogonal drawing* is a common drawing paradigm for visualizing planar graphs in numerous applications, ranging from VLSI circuit design [29, 42], architectural floor-planning [30], UML diagram design [39], and a popular visualization technique used in network layout systems, e.g. see [9, 16, 43]. For readability and aesthetics of such a drawing, traditional orthogonal layout algorithms [3, 27, 33] try to minimize the number of bends on the edges in the drawing. However, a recent empirical study by Kieffer *et al.* [28] concluded that orthogonal layouts generated by traditional automatic algorithms that focus primarily on bend and area minimization lack some aesthetic qualities compared to manual drawings produced by humans. Their study instead suggests that, like in several of the works by Lombardi, humans prefer drawings with linear “flow” that connect chains of adjacent vertices. Our specific interest in this paper is to study such “Lombardi flow” for orthogonal graph drawings.

Related Work. In a seminal work, Tamassia [40] showed that for a *plane graph* (i.e., a planar graph with a fixed planar embedding) with the maximum degree 4, an orthogonal drawing with the minimal number of edge bends can be computed in $O(n^2 \log n)$ time, by solving a minimum-cost maximum-flow problem. This time complexity was later reduced to $O(n^{7/4} \sqrt{\log n})$ by Garg and Tamassia [22], to $O(n^{1.5})$ by Cornelsen and Karrenbauer [10], and to linear time for plane graphs with maximum degree 3 by Rahman and Nishizeki [35]. In contrast, if the embedding is not given, the problem becomes NP-hard [21]. Bertolazzi *et al.* [5] proposed a branch-and-bound approach for minimizing the number of bends over all possible embeddings. There have also been some work on restricting the number of bends locally, e.g., on minimizing the maximum number of bends per edge [6, 7] in an orthogonal drawing instead of the total number of bends. Biedl and Kant [6] gave a linear-time algorithm for constructing an orthogonal drawing with at most two bends per edge for any graph with degree at most four except for the octahedron. Orthogonal drawings of graphs with vertices of degree higher

than four have also been studied [5, 18, 41]. Bläsius *et al.* [7] give a polynomial-time algorithm to decide whether a plane graph has an orthogonal drawing with at most one bend per edge. There has also been some research on *rectangular drawings*, where each face is rectangle [1, 36]. This is not the same as a minimum-segment drawing, though, since not every planar graph with maximum degree 4 admits a rectangular drawing, and even when such a drawing exists, it does not necessarily give a drawing with the minimum number of segments containing all vertices, e.g., see 7 in the appendix. In contrast, minimizing the number of non-orthogonal segments in a drawing of a graph has been previously addressed, in the context of straight-line drawings [11, 14, 15, 26, 32, 37]. Minimizing the number of circular arcs to draw planar graphs have also been studied [4, 25, 38]. However to the best of our knowledge, this problem has never been studied before in the context of orthogonal drawings.

Our Results. We study two problems for orthogonal drawings of planar graphs. These problems attempt to capture and formalize the concept of “Lombardi flow:”

1. A *minimum segment orthogonal drawing*, or *MSO-drawing*, of a planar graph G is an orthogonal drawing of G with the minimum number of segments. Given a combinatorial embedding of a degree-4 planar graph, G , the MSO-drawing problem is to produce such a drawing of G .
2. A *minimum segment cover orthogonal drawing*, or *MSCO-drawing*, of G is one with the smallest set of segments covering all vertices of G . Given a combinatorial embedding of a degree-4 planar graph, G , the MSCO-drawing problem is to produce such a drawing of G .

In this paper, we present the following results.

- We provide an interesting modification to the network-flow algorithm by Tamassia [40] to give a polynomial-time algorithm for the MSO-drawing problem.
- We show that the MSCO-drawing problem is NP-hard even for plane graphs with maximum degree 3.
- For trees and series-parallel graphs with maximum degree 3, we give polynomial-time algorithms for upward orthogonal drawings with the minimum number of segments covering the vertices. This solves the upward versions of the MSCO-drawing problem in polynomial time for these types of graphs.

2 Preliminaries

An *orthogonal drawing* of a planar graph, G , is a planar drawing of G such that each vertex is drawn as a point and each edge is drawn as a orthogonal (axis-aligned) path between the corresponding points. For an orthogonal drawing, it is generally assumed that the input graph is planar with maximum degree 4, and that for each vertex, at most one edge is incident in its four orthogonal *ports*: left,

right, top, bottom. Although models incorporating vertices with degree more than 4 were introduced later (e.g., by Fößmeier and Kaufmann [18]), in this article we assume that the input graph is always a planar graph with maximum degree 4. A *maximal line-segment* of an orthogonal drawing, Γ , is a set of line-segments (all horizontal or all vertical) l_1, l_2, \dots, l_k , such that k is as large as possible so that, for $i \in 1, 2, \dots, k-1$, l_i and l_{i+1} have a common end point. In this paper we simply use the term *segment* to refer to a maximal line-segment.

Tamassia [40] showed that an orthogonal drawing, Γ , of a biconnected plane graph, G , can be described by augmenting the embedding of G with the angles at the bends (*bend angles*) and the angles between pairs of consecutive edges around the vertices of G (*vertex angles*). Specifically, an *angle assignment* is a mapping from the set $\{\pi/2, \pi, 3\pi/2\}$ to the angles of G , where each angle is assigned exactly one value. An angle assignment of G can precisely describe the shape of Γ , although it does not specify edge lengths. Given an angle assignment Φ , one can test if Φ corresponds to an orthogonal drawing of G by Lemma 1, which is implied from [40]; also see [1].

Lemma 1 (Tamassia [40]). *An angle assignment Φ for a plane graph G corresponds to an orthogonal drawing of G if and only if Φ satisfies the following conditions (P_1 – P_2).*

- (P_1) *The sum of the assigned angles around each vertex v in G is 2π .*
- (P_2) *The number of $\pi/2$ angle around each face f is exactly 4 more (resp. 4 less) than the number of $3\pi/2$ angle around f , if f is an inner face (resp. the outer face).*

Given an angle assignment Φ satisfying (P_1 – P_2), one can obtain an orthogonal drawing of G (i.e., the exact coordinates for the vertices) in linear time.

3 Minimum Segment Orthogonal Drawing

In this section, we address the problem of finding an orthogonal drawing of an embedded planar graph, G , with maximum degree 4 using a minimum number of segments. We call such a drawing a *minimum-segment orthogonal drawing*, or an *MSO-drawing*. We show that by a modification and generalization of a the network-flow algorithm by Tamassia [40], we can give a polynomial-time algorithm to find an MSO-drawing of G . In addition, this generalization also enables an interactive trade-off adjustment in the orthogonal drawing between bends at a degree-2 vertex and edge bends, if that is desired.

We first briefly review the minimum-cost flow-network formulation of Tamassia [40] for computing minimum-bend orthogonal drawings of plane graphs. We then describe how this network can be modified and generalized to compute drawings with the minimum number of segments. We first assume that the input graph is connected and the degree of each vertex in the input graph is at least 2. In Section 3 we consider a general planar graph with maximum degree 4.

Network Flow-Formulation by Tamassia. Given a plane graph G with maximum degree 4, the algorithm by Tamassia [40] constructs a flow network H in order to find an orthogonal layout with the minimum total number of bends on the edges. In Tamassia’s network, H , there are *boundary-vertices*, representing the vertices of G , and *face-vertices*, representing the faces of G ; see Fig. 2(a). The edges of H are the bidirectional edges of the dual graph of G (dashed grey edges in Fig. 2(a), called *dual edges*) and the edges from each boundary-vertex to its incident face-vertices (solid grey edges, called *angle edges*). Each boundary-vertex, v , is a source that produces 4 units. Each face-vertex for a face f is a sink that consumes either $2 \deg(f) - 4$ units (for inner faces) or $2 \deg(f) + 4$ units (for the outer face), where $\deg(f)$ is the number of vertices on the face f . The cost of an edge is 1 unit if it connects two face-vertices (i.e., a dual edge), and 0 otherwise. A min-cost max-flow in this network corresponds to an orthogonal drawing of G , as follows. A flow of $t \in \{1, 2, 3\}$ units from a boundary-vertex to a face-vertex determines a $t\pi/2$ assignment to the corresponding angle in G . A flow of t units through some dual edge corresponds to t bends in the corresponding edge of G ; see Fig. 2(b). Using this minimum-cost flow-network, Tamassia [40] gave an $O(n^2 \log n)$ -time algorithm for orthogonal drawing with minimum number of bends. Cornelsen and Karrenbauer [10] used the same network but improved the running time to $O(n^{1.5})$ with a faster min-cost max-flow algorithm for this planar network.

Flow Formulation for Minimum-Segment Orthogonal Drawings. We modify and generalize the flow network by Tamassia to obtain an orthogonal drawing with a minimum number of segments. In our modification, we change the consumption and production of the flows in the boundary and face vertices, modify some edge directions and costs, and interpret the flows on the edges slightly differently.

In our modified network, we again have the *boundary-vertices*, *face-vertices* and the bidirectional *dual edges*. However instead of the unidirectional angle edges from each boundary-vertices to its incident face-vertices, we have bidirectional *angle edges* between each boundary-vertices and its incident face-vertices; see Fig. 3(a). Each face-vertex for an inner face f is a source with a production of 4 units, while the face-vertex for the outer face is a sink with a consumption of 4 units. Each boundary-vertex for a vertex v with degree 3 or 4 in G is a sink with a consumption of 2, or 4 units, respectively, while boundary-vertices corresponding to vertices of degree 2 is neither a source nor a sink. Thus the consumption of a boundary-vertex for a vertex v is $2 \deg(v) - 4$, where $\deg(v)$ denotes the degree of v in G . We set the capacity of flow on each direction of an angle edge to be 1 unit, while the dual edges are uncapacitated in both directions. Finally, the cost of an edge is c_1 if it is a dual edge, and c_2 if it is an angle edge, for some constant c_1, c_2 . We discuss below how to make suitable assignments to c_1 and c_2 to obtain a minimum-segment orthogonal drawing.

We interpret the flow on different edges of H to obtain an angle assignment for G as follows. For each (bidirectional) angle edge e , a flow of 0 unit gives a straight (π) angle assignment to the corresponding angle, while a flow of 1 unit

from the face-vertex to the boundary vertex (resp. from the boundary vertex to the face-vertex) in e gives a $\pi/2$ (resp. $3\pi/2$) assignment to the angle. A flow of t units through each dual edge again corresponds to t bends in the corresponding edge of G . We first show that a valid flow in the modified flow network H corresponds to a valid angle assignment for an orthogonal layout of G .

Lemma 2. *Each valid assignment of flows on the edges of H corresponds to an angle assignment for G , satisfying Conditions (P_1-P_2) of Lemma 1, and vice versa.*

Proof. Consider first a valid flow \mathcal{F} in H . We show that this gives a desired angle assignment to G . For each vertex v with degree 4 in G , its corresponding boundary-vertex has four incident angle edges and a consumption of 4 unit. Thus the boundary-vertex has to receive 1 unit of flow in each angle, giving a $\pi/2$ assignment to each angle around v . Similarly, a boundary-vertex corresponding to a vertex of degree 3 has to receive 1 unit of flow from exactly two of its three incident angle edges. Thus, each degree-3 vertex has two $\pi/2$ and one π assignment around itself. Finally consider the boundary-vertex v for each degree-2 vertex. Either there is no flow in either of the angle edges incident to v (which gives two π angles around the degree-2 vertex), or v receives 1 unit of flow in one incident edge and gives back the flow in the other (which gives a $\pi/2$ and a $3\pi/2$ assignments to the two angles). Thus in all cases, for each vertex of G , the summation of the assigned angles around it is 2π , satisfying Conditions P_1 .

For each face-vertex, each outgoing flow (of 1 unit) corresponds to a $\pi/2$ angle (on a vertex or an edge bend), and each incoming flow (of 1 unit) corresponds to a $3\pi/2$ angle. Since each face-vertex corresponding to an inner face is a source with a production of 4 units, and the face-vertex for the outer face is a sink with a consumption of 4 units, Condition P_2 follows.

Conversely for each valid angle assignment Φ on G , we can construct a valid flow assignment on the edges of H as follows. By Condition P_1 , for each vertex of degree 4, all four of its incident angles are $\pi/2$, thus the corresponding boundary-vertex receives a total of 4 units of flow as necessary. Similarly, for each degree-3 vertex, exactly two of its incident angles is $\pi/2$, providing the sources of the necessary 2 units of incoming flow for the boundary-vertex. A degree-2 vertex can either have two incident π angles, or a $\pi/2$ and a $3\pi/2$ angles. In the first case, the boundary-vertex has no incoming or outgoing flow, while in the second case, it has equal (1 unit) incoming and outgoing flow. Again By Condition P_2 , the face-vertex for each inner face sends 4 more unit of flow than what it receives, while the face-vertex for the outer face receives 4 more unit of flows than what it sends. \square

We now discuss suitable values of the edge costs c_1, c_2 . Let \mathcal{F} be a valid flow in H , and let Γ be the corresponding orthogonal layout of G . Note that each edge bend incurs a cost of c_1 , while each $\pi/2$ and $3\pi/2$ angle assignment around a vertex in Γ incurs a cost of c_2 in \mathcal{F} . In any angle assignment for G , there are exactly four $\pi/2$ angles around each degree-4 vertex, and there are exactly two $\pi/2$ angles around each degree-3 vertex in G . On the other hand,

for each degree-2 vertex, there are either two π angles around it, or there is a $\pi/2$ and a $3\pi/2$ angle around it. Call a degree-2 vertex a *vertex-bend* in Φ if it has a $\pi/2$ and a $3\pi/2$ angle around it. Thus the total cost in \mathcal{F} is $c_1(4\#(V_4) + 3\#(V_3) + 2\#(B_v)) + c_2\#(B_e)$, where $\#(V_4)$, $\#(V_3)$, $\#(B_v)$, $\#(B_e)$ denotes the number of degree-4 vertices in G , the number of degree-3 vertices, the number of vertex bends in Γ , and the number of edge-bends in Γ , respectively. Minimizing the cost in this network for different values of c_1 and c_2 optimizes different properties of the resulting orthogonal layout. (One might verify that $c_1 = 1$, $c_2 = 0$ gives an orthogonal layout with minimum number of total edge bends, similar to Tamassia [40]). Lemma 3 gives a suitable value for c_1 , c_2 to minimize the number of segments in the corresponding orthogonal layout.

Lemma 3. *A minimum-cost flow in H for the values of $c_1 = 1/2$, $c_2 = 1$ gives an angle assignment of G corresponding to a minimum segment drawing of G .*

Proof. Let \mathcal{F} be a valid flow in H and let Γ be the corresponding orthogonal layout of G . Let $\#(V_4)$, $\#(V_3)$, $\#(B_v)$, $\#(B_e)$ denote the number of degree-4 vertices in G , the number of degree-3 vertices, the number of vertex bends in Γ , and the number of edge-bends in Γ , respectively. We count the number of segments in Γ . Each segment in Γ has two end points, and each end-point of a segment in Γ is a degree-3 vertex or a vertex-bend or an edge-bend. Conversely, each vertex or edge bend corresponds to the end point of exactly two segments, while each degree-3 vertex corresponds to the end point of exactly one segment. Thus the number of segments $\#(s)$ in Γ is $(\#(V_3) + 2\#(B_v) + 2\#(B_e))/2$. Now with $c_1 = 1/2$, $c_2 = 1$, the cost function of H is (from the previous paragraph): $(4\#(V_4) + 3\#(V_3) + 2\#(B_v))/2 + \#(B_e) = (2\#(V_4) + \#(V_3)) + (\#(V_3) + 2\#(B_v) + 2\#(B_e))/2 = k + \#(s)$, where $(2\#(V_4) + \#(V_3)) = k$ is a constant for any orthogonal drawing Γ of G . Hence for $c_1 = 1/2$, $c_2 = 1$, minimizing the cost function also minimizes the number of segments in Γ . \square

Note that H is a planar network with size linear to the number of vertices in G . Cornelsen and Karrenbauer [10] showed that if the total cost is linear, then a minimum-cost flow on a planar network can be computed in $O(n^{1.5} \log n)$ time. Thus summarizing the results in this section, we have the following theorem.

Theorem 1. *For an embedded connected n -vertex planar graph, G , with maximum degree 4 and no vertex of degree 1, a minimum segment orthogonal drawing of G can be computed in $O(n^{1.5} \log n)$ time.*

Computing Exact Drawing: Aligning Multiple Segments Together. In Section 3, we show how to compute an orthogonal representation for a planar graph by computing the angle assignment for the angles around each vertex and by specifying the edge bends. This is a topological description of a topologically equivalent class of orthogonal drawings, but such an orthogonal drawing is dimensionless. In order to create an exact orthogonal drawing from such a representation, Tamassia [40] first augments the layout into a rectangular layout by decomposing each non-rectangular face into rectangles with the introduction

of dummy edges and vertices. Then integer lengths are assigned to the sides of each rectangular face to obtain the exact coordinate.

As an additional extension to Tamassia’s algorithm, we give here an algorithm to augment an orthogonal representation into a rectangular layout such that the maximum number of pairs of segments align with each other, thereby extending the “visual flow” across faces in a drawing. Let Γ be an orthogonal layout of a plane graph G with the maximum degree 4, and let Γ^* be rectangular layout containing Γ . A pair of vertices v_1, v_2 are said to be *aligned together* if they belong to the same maximal line-segment in Γ^* , but do not belong to the same maximal line-segment in Γ . and both l_1, l_2 belong to the same maximal line-segment in Γ^* . Formally we address the following problem:

- **(Augmentation):** Given an orthogonal layout Γ of a plane graph G with the maximum degree 4, construct a rectangular layout Γ^* such that Γ is contained in Γ^* and the maximum number of pairs of vertices from G are aligned together in Γ^* .

We solve the Augmentation problem by computing a maximum matching for a bipartite graph constructed for each non-rectangular face of Γ independently. For each face f we construct a bipartite graph $B(f)$ whose edges represent potential pairs of vertices which can be aligned together inside f . We need to select a maximum matching on $B(f)$ in such a way that all the pairs of vertices corresponding to these matching edges can be aligned together simultaneously. In Appendix A we show that such a matching can be computed in polynomial time, using the Hopcroft-Karp algorithm [24] for example. Doing this for each non-rectangular faces in Γ gives an augmentation Γ^* of Γ into a rectangular layout with the maximum pairs of vertices aligned together; see Appendix A.

Theorem 2. *The Augmentation problem can be solved in polynomial time.*

General Planar Graphs with Maximum Degree 4. In Section 3 we considered planar graphs with maximum degree 4 and with no degree-1 vertex. Here we consider all planar graphs with the maximum degree 4. Let G be planar graph with the maximum degree 4. By repetitively deleting vertices of degree 1 from G , we get a subgraph G' of G with no vertex of degree 1 (possibly with a single vertex, in which case G is a tree). Call G' the *closure* of G . Call the vertices of G' with at least one adjacent vertex in G , not belonging to G' the *connection vertices* of G . From G , delete all vertices of G' . Each connected component of the resulting graph is then a tree. Given a general planar graph G with the maximum degree 4, we therefore draw the closure of G using the algorithm in Section 3, and attach the remaining tree parts to the connection vertices. For drawing a rooted tree T (rooted at a connection vertex) with the maximum degree 4, we first draw an arbitrary root-to-leaf path in a single segment. We then repetitively take a shortest path from a leaf to vertex already drawn, and draw the path in a segment to complete a minimum segment drawing of T . Finally we attach the drawing of each tree part to the drawing of the closure of G at the corresponding

connection vertex to get a minimum segment drawing for the embedding given by the drawing; see Fig. 9(b). The details are in Appendix A.

Theorem 3. *For an n -vertex planar graph, G , with maximum degree 4, an orthogonal drawing Γ of G can be obtained in polynomial time such that Γ is a minimum segment orthogonal drawing for the embedding of G in Γ .*

4 Minimum Segment Cover Orthogonal Drawing

Let Γ be an orthogonal drawing for a planar graph, G , with maximum degree 4. A set of segments S in Γ is said to *cover* G , if each vertex in G is on at least one of the segments in S . The *segment-cover number* of Γ is the minimum cardinality of a set of segments covering G . Given a planar graph G , with maximum degree 4, a *minimum segment cover orthogonal drawing* or *MSCO-drawing* of G is an orthogonal drawing that has the minimum cardinality of a set of segments covering G . The number of segments in such an optimal set is also called the *segment-cover number* of G .

NP-Hardness. We begin our discussion in this section by showing that finding a minimum segment cover orthogonal drawing is NP-hard, even for planar graphs with maximum degree 3.

Theorem 4. *Finding a minimum segment cover orthogonal drawing for a planar graph G is NP-hard, even if G is a planar graph with maximum degree 3.*

Proof. We prove the NP-hardness of the problem by a reduction from the *Hamiltonian path problem* [20]; please see the appendix for details. \square

Algorithm for Series-Parallel Graphs with Maximum Degree 3. A *series-parallel* graph is a graph G that has two terminals s and t , and either G is an edge (s, t) , or it has been constructed in one of the following ways: (1) (Parallel combination) G consists of two or more series-parallel graphs that all have the terminals s and t . (2) (Combination in series) G consists of two or more series-parallel graphs G_1, G_2, \dots, G_f such that each G_i , $i \in \{1, 2, \dots, f\}$ has terminals $\{s_i, t_i\}$ with terminals $t_i = s_{i+1}$ for $i \in \{1, 2, \dots, f\}$ and $s_1 = s$, $t_f = t$.

The recursive construction of a series-parallel graph is represented by an *SPQ-tree*, which is a rooted tree defined as follows. An SPQ-tree \mathcal{T} for a series-parallel graph G contains at most three types of nodes: *S*-, *P*-, and *Q*-nodes. Informally each edge of G corresponds to a *Q*-node, while an *S*-node and a *P*-node denote a series and a parallel combination, respectively. If G contains a single edge, then \mathcal{T} consists of a single *Q*-node. Otherwise if G is obtained by a parallel (resp. series) combination of two or more series-parallel graphs G_1, G_2, \dots, G_f , then \mathcal{T} is rooted at a *P*-node (resp. *S*-node) with the SPQ-trees for G_1, G_2, \dots, G_f as children subtrees of the root.

In this section we consider series-parallel graphs with maximum degree 3. For each such graph, Lemma 4 gives some properties of its SPQ-tree, which is implied from [37].

Lemma 4. [37] *Let G be a series-parallel graph with the maximum degree 3 and let \mathcal{T} be its SPQ-tree. Then the following two conditions hold:*

- *For each S -node x of \mathcal{T} with children nodes x_1, x_2, \dots, x_f , (i) each x_i , $i \in \{1, 2, \dots, f\}$ is either a P -node or a Q -node, (ii) both x_1 and x_f are Q -nodes, and (iii) if x_i is a P -node for some $i \in \{2, \dots, f-1\}$, then both x_{i-1} and x_{i+1} are Q -nodes.*
- *Each non-root P -node has exactly two children and a root P -node (if any) has at most three children. At most one of the children of a P -node is a Q -node, and the remaining children are S -nodes.*

A polyline l is y -monotone if for any horizontal line h , the intersection $l \cap h$ is a connected line-segment (possibly empty or a single point). Let \mathcal{T} be the SPQ-tree of a series-parallel G such that the terminals for the root node of \mathcal{T} are s and t . Then an orthogonal drawing Γ of G is *upward* if any path from s to t in G is drawn as a y -monotone polyline in Γ .

Theorem 5. *Let G be a series-parallel graph with the maximum degree 3 with the SPQ-tree \mathcal{T} and let $\#(P^*)$ be the number of P -nodes in \mathcal{T} with at least two S -nodes as children. If the root of \mathcal{T} is a P -node with three S -nodes as children, then $\#(P^*) + 2$ segments are necessary and sufficient to cover all vertices of G in any upward orthogonal drawing of G ; otherwise $\#(P^*) + 1$ segments are necessary and sufficient.*

We prove Theorem 5 by constructing an algorithm that obtains an upward orthogonal drawing of G with the desired segment cover number, and then proving that this bound is also necessary. Assume first that each P -node in the SPQ-tree \mathcal{T} for G (including the root, if it is a P -node) has exactly 2 children.

Lemma 5. *Let G be a series-parallel graph with the maximum degree 3 with the SPQ-tree \mathcal{T} where each P -node has exactly 2 children. Let $\#(P^*)$ be the number of P -nodes in \mathcal{T} with two S -nodes as children. Then there is an upward orthogonal drawing of G with $\#(P^*) + 1$ vertical segments covering all the vertices.*

Proof. We give an inductive construction of an upward orthogonal drawing of G . If G contains only a single edge (i.e., the root r of \mathcal{T} is a Q -node), then we draw G with a single vertical line segment. Otherwise, if r is either an S -node or a P -node. If r is an S -node with children x_1, \dots, x_f , then let G_1, \dots, G_f be the series-parallel graphs corresponding to the subtrees of \mathcal{T} rooted at x_1, \dots, x_f , respectively. For $i \in \{1, 2, \dots, f\}$, let s_i, t_i be the two terminals of G_i and let $\#(P_i^*)$ denote the number of P -nodes with two children S -nodes in the subtree rooted at x_i . We then recursively draw upward orthogonal drawing $\Gamma(x_i)$ for G_i with $\#(P_i^*) + 1$ vertical segments covering all the vertices of G_i , for $i \in \{1, \dots, f\}$. We then obtain an upward orthogonal drawing of G by placing the drawings $\Gamma(x_1), \Gamma(x_2), \dots, \Gamma(x_f)$ one after another from top to bottom and by identifying t_i with s_{i+1} for $i \in \{1, 2, \dots, f-1\}$; see Fig. 5(a). This identification of vertices are possible since the drawings $\Gamma(x_i)$'s are upward (hence s_i and t_i are at the topmost and bottommost y -coordinates) for each $i \in \{1, \dots, f\}$. Furthermore,

by Lemma 4, for each P -node child x_i , both x_{i-1} and x_{i+1} are Q -nodes (and hence are drawn as a vertical line segment). Thus there is no edge overlap in the resulting drawing. Since each drawing $\Gamma(x_i)$ has $\#(P_i^*) + 1$ vertical segments covering all the vertices of G_i , and for $i \in \{2, \dots, f\}$, $\Gamma(x_i)$ shares exactly one of these vertical segments with $\Gamma(x_{i-1})$, the total vertical segments covering all the vertices of G is $(\#(P_1^*) + 1) + (\#(P_2^*)) + \dots + (\#(P_f^*)) = \#(P^*) + 1$.

Finally if the root r of \mathcal{T} is a P -node with children x_1 and x_2 , then let G_1, G_2 be the two series-parallel graphs corresponding to the two subtrees of \mathcal{T} rooted at x_1 and x_2 , respectively. For $i \in \{1, 2\}$, let s_i, t_i be the two terminals of G_i and let $\#(P_i^*)$ denote the number of P -nodes with two children S -nodes in the subtree rooted at x_i . We recursively obtain upward orthogonal drawings $\Gamma(x_1)$ and $\Gamma(x_2)$ of G_1, G_2 , respectively. Let x_1 is an S node. Then we place the two drawings $\Gamma(x_1), \Gamma(x_2)$ side-by-side and identify s_1, s_2 and t_1, t_2 by making two bends at the very top edge and the very bottom edge in $\Gamma(x_2)$; see Fig. 5(b). There is only a single edge at the top and the bottom since x_2 is either a Q -node or an S -node, and if x_2 is an S -node, then by Lemma 4, its topmost and bottommost part contain a single vertical segment for its children Q -nodes. If x_2 is an S -node, then $\#(P^*) = \#(P_1^*) + \#(P_2^*) + 1$, and the total number of vertical segments covering all vertices of G is $(\#(P_1^*) + 1) + (\#(P_2^*) + 1) = \#(P^*) + 1$. Otherwise, The segment between the two bend points in $\Gamma(x_2)$ does not contain any vertex, and the total number of vertical segments covering all vertices of G is $(\#(P_1^*) + 1) = \#(P^*) + 1$. \square

Fig. 6 illustrates an example of a drawing obtained from the algorithm in the proof of Lemma 5. We are now ready to prove Theorem 5.

Proof of Theorem 5. If the root r of \mathcal{T} is not a P -node, or if it has only two children, then we find an upward orthogonal drawing of G by the algorithm described in the proof of Lemma 5. Otherwise r is a P -node with its three children x_1, x_2, x_3 . \mathcal{T}_1 is the SPQ-tree obtained after deleting the subtree rooted at x_3 from \mathcal{T} . Also \mathcal{T}_2 is the SPQ-tree obtained after deleting the subtrees rooted at x_1, x_2 from \mathcal{T} . Let G_1, G_2 be the series-parallel graphs corresponding to these two SPQ-trees, respectively. We find upward orthogonal drawings Γ_1, Γ_2 for G_1, G_2 , place them side-by-side and identify their two pairs of terminals by making bends at the very top and the very bottom segments in Γ_2 . This gives an upward orthogonal drawing of G with the desired segment-cover number.

To see that the drawing achieves the optimal segment-cover number, observe the following. Let x_p be an arbitrary non-root P -node in \mathcal{T} with two S -node children x_1, x_2 . Let s and t be the two terminals corresponding to x_p . Then deleting $\{s, t\}$ splits G into exactly three components, two containing vertices of the two series parallel graphs G_1, G_2 corresponding to the subtrees rooted at x_1 and x_2 , respectively, and the third containing vertices from neither G_1 nor G_2 . In any upward orthogonal drawing of G , both s and t (or their incident edges) will be the end point of at least one segments that contains vertices of G . By Lemma 4, no two P -nodes share any terminals. Thus if $\#(P_n^*)$ be the number of non-root P -nodes, then there are at least $2\#(P_n^*)$ end-points of segments containing vertices, and thus at least $\#(P_n^*)$ number of segments containing

vertices in any upward orthogonal drawing. Finally If the the root node r is a P node, then since the drawing is upward, none of the graph corresponding to its each child S -node and its descendants can share a segment covering vertices, hence the result follows. \square

An upward drawing of a series-parallel graph with the maximum degree 3 is not always an MSCO-drawing; see Fig. 6(d). However for any outer-planar graph (which is a series-parallel graph), each P -node in the SPQ-tree has exactly one Q -node and one S -node as children. Since a segment is necessary to cover the vertices of any graph, this observation gives Corollary 1 from Theorem 5.

Corollary 1. *Let G be a series-parallel graph with the maximum degree 3 with the SPQ-tree \mathcal{T} and let $\#(P^*)$ be the number of P -nodes in \mathcal{T} with no Q -node as a child. Then $\#(P^*) + 1$ segments are sometimes necessary and always sufficient to cover all vertices of G in any orthogonal drawing of G .*

Algorithm for Rooted Trees. Our algorithm for rooted trees in Section 3 gives an algorithm for obtaining an orthogonal drawing for trees with the optimal segment-cover number, since the end point of each segment in the drawing is either a degree-1 or a degree-3 vertex; see Fig. 4(a). However this drawing is not necessarily upward (i.e., where each root-to-leaf path is y -monotone). Note that for an upward orthogonal drawing of a rooted tree, the maximum degree of a vertex is at most 3. We can obtain an upward drawing for rooted trees with maximum degree-3 by using our algorithm from Section 4. Let T be a rooted tree. Take two copies of T , call them T_1, T_2 . Let r_1, r_2 be their respective roots. Clearly there is a trivial isomorphism between T_1 and T_2 . Now identify each leaf of T_1 with the corresponding leaf in T_2 . The resulting graph is then a series-parallel graph with the maximum degree 3. An upward orthogonal drawing of this graph using the algorithm from Section 4 also gives an upward drawing of T (as a drawing of T_1 and T_2) with optimal segment-cover number; see Fig. 4(b).

5 Conclusion and Future Work

We have provided a characterization of “Lombardi flow” in orthogonal drawings in terms of minimizing the total number of segments in such a drawing or in the number of segments covering all the vertices in such a drawing. Given that this latter version of the problem is NP-hard, as we have shown, an interesting direction for future work includes possible approximation algorithms for general degree-4 planar graphs, as well as the study of other graph classes that admit polynomial-time solutions.

Acknowledgements. This article reports on work supported by the Defense Advanced Research Projects Agency under agreement no. AFRL FA8750-15-2-0092. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This work was also supported in part by the U.S. National Science Foundation under grants 1228639 and 1526631. We would like to thank Timothy Johnson and Michael Bekos for several helpful discussions related to the topics of this paper.

References

1. Alam, M.J., Kobourov, S.G., Mondal, D.: Orthogonal layout with optimal face complexity. In: International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'16). Lecture Notes in Computer Science, vol. 9587, pp. 121–133. Springer (2016)
2. Alam, M.J., Bekos, M.A., Kaufmann, M., Kindermann, P., Kobourov, S.G., Wolff, A.: Smooth orthogonal drawings of planar graphs. In: Latin American Symposium on Theoretical Informatics (LATIN'14). Lecture Notes in Computer Science, vol. 8392, pp. 144–155. Springer (2014)
3. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. The MIT Press, 3rd edn. (2009)
4. Bekos, M.A., Raftopoulou, C.N.: On a conjecture of lovász on circle-representations of simple 4-regular planar graphs. *Journal of Computational Geometry (JoCG)* 6(1), 1–20 (2015)
5. Bertolazzi, P., Battista, G.D., Didimo, W.: Computing orthogonal drawings with the minimum number of bends. *IEEE Transactions on Computers* 49(8), 826–840 (2000)
6. Biedl, T.C., Kant, G.: A better heuristic for orthogonal graph drawings. *Computational Geometry* 9(3), 159–180 (1998)
7. Bläsius, T., Krug, M., Rutter, I., Wagner, D.: Orthogonal graph drawing with flexibility constraints. *Algorithmica* 68(4), 859–885 (2014)
8. Chernobelskiy, R., Cunningham, K.I., Goodrich, M.T., Kobourov, S.G., Trott, L.: Force-directed Lombardi-style graph drawing. In: Graph Drawing (GD'11). Lecture Notes in Computer Science, vol. 7034, pp. 320–331. Springer (2012)
9. Chimani, M., Gutwenger, C., Jünger, M., Klau, G., Klein, K., Mutzel, P.: The open graph drawing framework. In: Handbook of Graph Drawing and Visualization, pp. 543–571. CRC Press (2013)
10. Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. *Journal of Graph Algorithms and Applications* 16(3), 635–650 (2012)
11. Dujmovic, V., Eppstein, D., Suderman, M., Wood, D.R.: Drawings of planar graphs with few slopes and segments. *Computational Geometry* 38(3), 194–212 (2007)
12. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Löffler, M.: Planar and poly-arc Lombardi drawings. In: Graph Drawing (GD'11). Lecture Notes in Computer Science, vol. 7034, pp. 308–319. Springer (2012)
13. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Nöllenburg, M.: Lombardi drawings of graphs. *Journal of Graph Algorithms and Applications* 16(1), 85–108 (2012)
14. Durocher, S., Mondal, D.: Drawing plane triangulations with few segments. In: Canadian Conference on Computational Geometry (CCCG'14). pp. 40–45 (2014)
15. Durocher, S., Mondal, D., Nishat, R.I., Whitesides, S.: A note on minimum-segment drawings of planar graphs. *Journal of Graph Algorithms and Applications (JGAA)* 17(3), 301–328 (2013)
16. Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C., Woodhull, G.: Graphviz - open source graph drawing tools. In: Symposium on Graph Drawing (GD). pp. 483–484 (2001)
17. Eppstein, D.: Planar Lombardi drawings for subcubic graphs. In: Graph Drawing (GD'12). Lecture Notes in Computer Science, vol. 7704, pp. 126–137. Springer (2013)

18. Fößmeier, U., Kaufmann, M.: Drawing high degree graphs with low bend numbers. In: Symposium on Graph Drawing (GD). pp. 254–266 (1995)
19. Garey, M.R., Johnson, D.S., Tarjan, R.E.: The planar hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing* 5(4), 704–714 (1976), <http://dx.doi.org/10.1137/0205049>
20. Garey, M.R., Johnson, D.S.: *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA (1990)
21. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing* 31(2), 601–625 (2001)
22. Garg, A., Tamassia, R.: A new minimum cost flow algorithm with applications to graph drawing. In: Graph Drawing (GD’96). Lecture Notes in Computer Science, vol. 1190, pp. 201–216. Springer (1997)
23. Hobbs, R.C., Richards, J.: Mark Lombardi: Global Networks. Independent Curators International (2003)
24. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing* 2(4), 225–231 (1973)
25. Hültschmidt, G., Kindermann, P., Meulemans, W., Schulz, A.: Drawing trees and triangulations with few geometric primitives. In: European Workshop on Computational Geometry (EuroCG’16) (2016)
26. Igamberdiev, A., Meulemans, W., Schulz, A.: Drawing planar cubic 3-connected graphs with few segments: Algorithms and experiments. In: Graph Drawing and Network Visualization (GD’15). Lecture Notes in Computer Science, vol. 9411, pp. 113–124. Springer (2015)
27. Kaufmann, M., Wagner, D.: *Drawing Graphs: Methods and Models*, Lecture Notes in Computer Science, vol. 2025. Springer-Verlag, London, UK (2001)
28. Kieffer, S., Dwyer, T., Marriott, K., Wybrow, M.: HOLA: human-like orthogonal network layout. *IEEE Transactions on Visualization and Computer Graphics* 22(1), 349–358 (2016)
29. Leiserson, C.E.: Area-efficient graph layouts (for VLSI). In: Symposium on Foundations of Computer Science (FOCS). pp. 270–281 (1980)
30. Liggett, R.S., Mitchell, W.J.: Optimal space planning in practice. *Computer-Aided Design* 13(5), 277–288 (1981)
31. Löffler, M., Nöllenburg, M.: Planar Lombardi drawings of outerpaths. In: Graph Drawing (GD’12). Lecture Notes in Computer Science, vol. 7704, pp. 561–562. Springer (2013)
32. Mondal, D., Nishat, R.I., Biswas, S., Rahman, M.S.: Minimum-segment convex drawings of 3-connected cubic plane graphs. *Journal of Combinatorial Optimization* 25(3), 460–480 (2013)
33. Nishizeki, T., Rahman, M.S.: *Planar Graph Drawing*. World Scientific, Singapore (2004)
34. Purchase, H.C., Hamer, J., Nöllenburg, M., Kobourov, S.G.: On the usability of Lombardi graph drawings. In: Graph Drawing (GD’12). Lecture Notes in Computer Science, vol. 7704, pp. 451–462. Springer (2013)
35. Rahman, M.S., Nishizeki, T.: Bend-minimum orthogonal drawings of plane 3-graphs. In: International Workshop Graph-Theoretic Concepts in Computer Science (WG’02). Lecture Notes in Computer Science, vol. 2573, pp. 367–378. Springer (2002)
36. Rahman, M.S., Nishizeki, T., Ghosh, S.: Rectangular drawings of planar graphs. *Journal of Algorithms* 50(1), 62–78 (2004)

37. Samee, M.A.H., Alam, M.J., Adnan, M.A., Rahman, M.S.: Minimum segment drawings of series-parallel graphs with the maximum degree three. In: Graph Drawing (GD'08). Lecture Notes in Computer Science, vol. 5417, pp. 408–419. Springer (2009)
38. Schulz, A.: Drawing graphs with few arcs. *Journal of Graph Algorithms and Applications (JGAA)* 19(1), 393–412 (2015)
39. Seemann, J.: Extending the sugiyama algorithm for drawing UML class diagrams: Towards automatic layout of object-oriented software diagrams. In: Graph Drawing (GD'97). Lecture Notes in Computer Science, vol. 1353, pp. 415–424. Springer (1997)
40. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing* 16(3), 421–444 (1987)
41. Tamassia, R., Battista, G.D., Batini, C.: Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics* 18(1), 61–79 (1988)
42. Valiant, L.G.: Universality considerations in VLSI circuits. *IEEE Transaction on Computers* 30(2), 135–140 (1981)
43. Wiese, R., Eiglsperger, M., Kaufmann, M.: yFiles visualization and automatic layout of graphs. In: Graph Drawing Software, pp. 173–191. Springer (2004)

A Figures and Proof Details

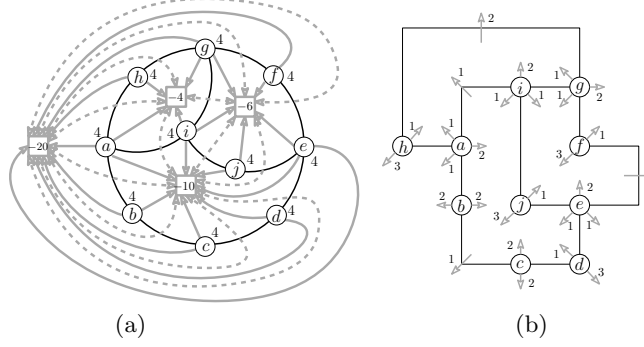


Fig. 2: (a) Construction of the flow network (with grey edges) by Tamassia [40] for a planar graph, G , with maximum degree 4 (black edges), (b) an orthogonal drawing of G and the corresponding flows in the network.

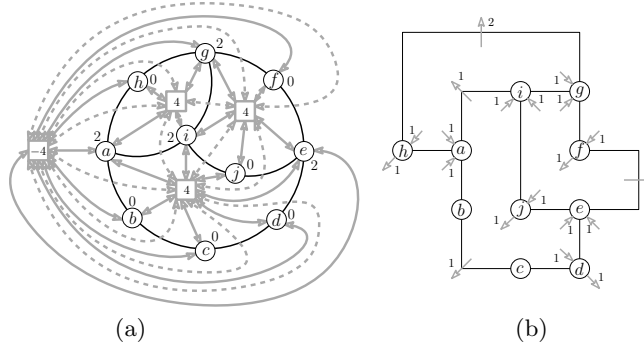


Fig. 3: (a) Construction of the modified flow network (with grey edges) for the graph G (black edges) from Fig. 2(a), (b) the drawing of G from Fig. 2(b) and the corresponding flows in the modified network.

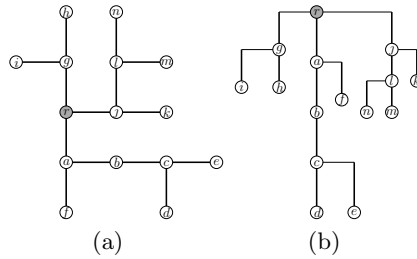


Fig. 4: Orthogonal drawings of trees: (a) MSO-drawing and MSCO-drawing, (b) upward drawing with the minimum segment-cover number.

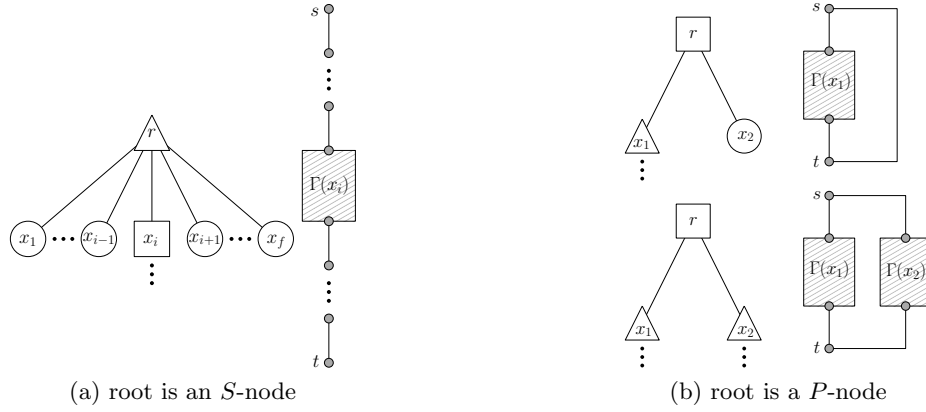


Fig. 5: Upward orthogonal drawing algorithm for a series-parallel graph. S -, P -, and Q -nodes are denoted by triangles, squares and circles, respectively.

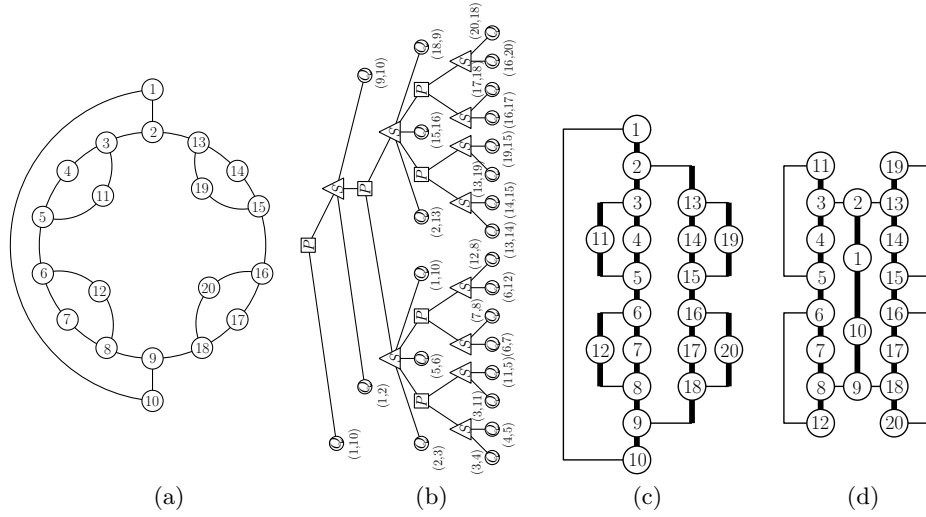


Fig. 6: (a) A series parallel graph G , (b) the SPQ-tree of G , (c) an upward drawing of G with the minimum segment-cover number, (d) an MSCO-drawing of G .

Proof of Theorem 2. Let f be a non-rectangular face in Γ . A vertex v on f is called *left-open* (resp. *right-open*, *top-open*, *bottom-open*) in f if the left (resp. right, top, bottom) port of v is inside f and is not incident to any edge in Γ . Note that a pair of vertices v_1, v_2 on f can align together only if one of them is left-open and the other is right open, or one of them is top open and the other bottom-open. We construct a bipartite graph $B(f)$ for f as follows. The vertices of $B(f)$ are all the vertices on f , which are left-open, or right-open, or top-open or bottom-open. The edges of $B(f)$ are between each pair of vertices (v_1, v_2) such that either (i) v_1 is left-open, v_2 is right-open (call them *horizontal pair*), or (ii) v_1 is top-open, v_2 is bottom-open (call them *vertical pair*). Each edge of $B(f)$ represents a potential pair of vertices which can be aligned together. However, we need to select a set of such pairs such that (i) each vertex is paired with at most one other, and (ii) the chosen pairs together can augment f to a set of rectangular faces. For the two conditions to hold, the edges corresponding to all pairs in f must form a maximum matching in $B(f)$ and it should be possible to place the edges for all horizontal pairs (resp. all vertical pairs) inside f without crossing. Two horizontal (resp. vertical) pairs are *in conflict* if both of them cannot be placed together inside f without crossing. A maximum matching in $B(f)$ is called *planar* if no two horizontal pairs and no two vertical pairs corresponding to these matching edges are in conflict. We have the following lemma.

Lemma 6. *Let M be a maximum matching of $B(f)$. Then a planar maximum matching of $B(f)$ can be constructed from M in polynomial time.*

Proof. If M contains no pair of edges in conflict, then we are done. Thus consider a pair of edges $(a, c), (b, d)$ from M are in conflict. Assume without loss of generality that a, b, c, d appear in this clockwise order around f . Then $M' = (M - \{(a, c), (b, d)\}) \cup \{(a, d), (b, c)\}$ is another maximum matching of $B(f)$ with one fewer pairs in conflict. A repetition of this procedure eliminates all pairs of M in conflict to obtain a planar maximum matching of $B(f)$. \square

Since finding a maximum matching in a bipartite graph can be done in polynomial time (e.g., using the Hopcroft-Karp algorithm [24]), by Lemma 6, one can find a planar maximum matching of $B(f)$ in polynomial time. Adding the edges corresponding to the matching might create crossing between horizontal and vertical edges. However we can introduce a dummy vertex at the point of the crossing to obtain a valid rectangular layout of f . Doing this for each non-rectangular faces in Γ gives an augmentation Γ^* of Γ into a rectangular layout. Since we maximized number of aligned pairs inside each face of Γ , Γ^* also gives a solution to the Augmentation problem. \square

Proof of Theorem 3. Given a general planar graph G with the maximum degree 4, we draw the closure of G using the algorithm in Section 3, and attach the remaining tree parts to the connection vertices.

Consider a rooted tree T with the maximum degree 4. We find a minimum segment drawing of T as follows. We first find an arbitrary root-to-leaf path and

draw the path in a single segment. Next we repetitively take a leaf v_l of T not already drawn, find a shortest path P_l from v_l to a vertex v_p already drawn and draw P_l as a single segment starting from a port (top, bottom, left or right) of v_p not already occupied by a different edge. If v_p is an odd-degree vertex in the already drawn graph, we place the segment for P_l in such a way that it shares an incident segment at v_p already drawn. The number of segments in such a drawing of T is exactly $(\#(V_1) + \#(V_3))/2$, where $\#(V_1)$ and $\#(V_3)$ are the number of degree-1 and degree-3 vertices in T .

Now we describe our algorithm for a general planar graph G with the maximum degree 4. We first draw the closure of G using the algorithm in Section 3. Next we draw each connected component (say T) of the graph obtained by deleting the closure of G from G using the algorithm describe in the previous paragraph. Note that T is a tree. We attach the drawing of T at the corresponding connection vertex at a port not already used. Again if a connection vertex has odd degree in the already drawn graph we place the drawing of T such that it shares a segment with the existing drawing.

The resulting drawing Γ gives an embedding of the input graph G . The drawing Γ is a minimum segments drawing for the embedding since the drawing of the closure minimized the vertex bends and edge bends, and the remaining parts of the graph has not vertex or edge bends. \square

Proof of Theorem 4. We prove the NP-hardness of the MSCO-drawing problem by a reduction from the *Hamiltonian path problem* [20], which asks whether a given graph G has a path that contains each vertex of G exactly once (i.e., a *Hamiltonian path*). The Hamiltonian path problem remains NP-complete for planar graphs with maximum degree 3 [19]. Let G be such a planar graph. We show that G has a Hamiltonian path if and only if its segment-cover number is 1.

Clearly, if G has an orthogonal drawing that contains a segment covering all the vertices, then the path of G corresponding to this segment is a Hamiltonian path. Conversely if G has a Hamiltonian path P , then we can find an orthogonal drawing of G with a single segment covering all the vertices as follows. Take a fixed planar embedding $\mathcal{E}(G)$ of G , draw P as a single segment (say a horizontal segment) and route the remaining edges as orthogonal polylines with necessary bends keeping the embedding $\mathcal{E}(G)$ fixed. This gives a valid orthogonal drawing since each vertex has a maximum degree 3, and hence from each vertex v , its incident edges on P are drawn to its left and right, while the only other edge (if any) is drawn to its top or bottom without any overlap. \square

B Examples for MSCO-Drawings

In this section we illustrate some examples of an orthogonal drawing obtained from the algorithm by Tamassia [40], along with an MSO-drawing, and an MSCO-drawing of the same graph.

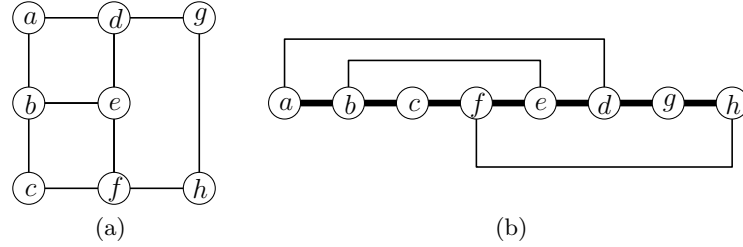


Fig. 7: (a) A drawing of a planar graph G using the algorithm by Tamassia [40], which is also a rectangular drawing, (b) an MSCO-drawing of G with only a single segment covering all the vertices.

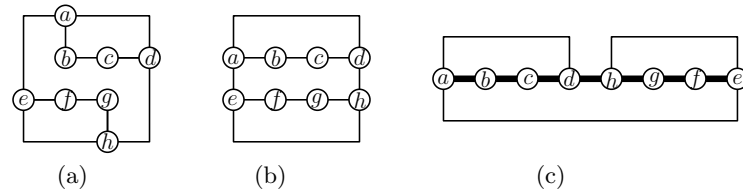


Fig. 8: (a) A drawing of a planar graph G using the algorithm by Tamassia [40], (b) an MSO-drawing of G , (c) an MSCO-drawing of G with only a single segment covering all the vertices.

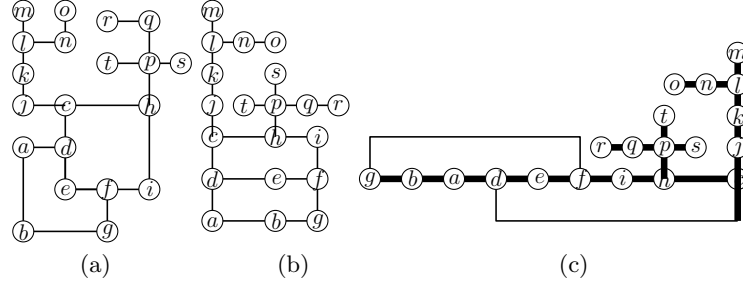


Fig. 9: (a) A drawing of a planar graph G using the algorithm by Tamassia [40], (b) an MSO-drawing of G , (c) an MSCO-drawing of G .

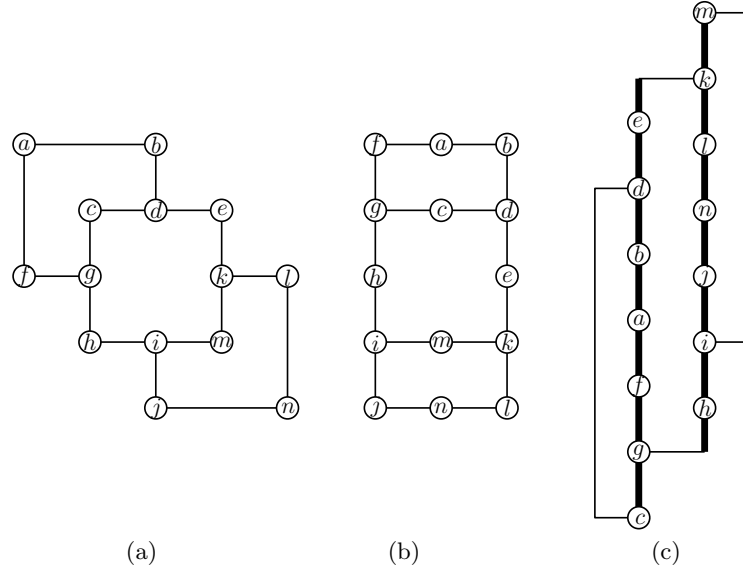


Fig. 10: (a) A drawing of a planar graph G using the algorithm by Tamassia [40], (b) an MSO-drawing of G , (c) an MSCO-drawing of G with only a single segment covering all the vertices.

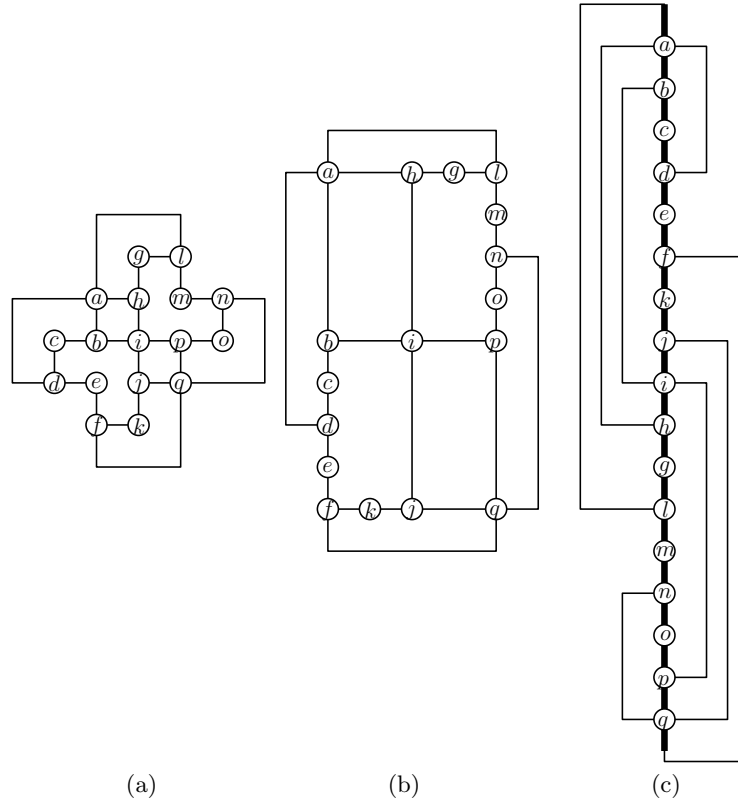


Fig. 11: (a) A drawing of a planar graph G using the algorithm by Tamassia [40], (b) an MSO-drawing of G , (c) an MSCO-drawing of G with only a single segment covering all the vertices.